DOCKET NO. 05-01-004
CLIENT NO.: UGSC01-05028
Customer No. 45113

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	:	SCOTT R. MCMULLAN, ET AL.
Serial No.	:	09/675,699
Filed	:	September 29, 2000
For	:	METHOD OF REAL-TIME BUSINESS COLLABORATION
Group No.	:	2165
Examiner	:	H. Mahmoudi

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

Sir:

The undersigned hereby certifies that the following documents:

1.   Appeal Brief (in triplicate);
2.   Check in the amount of $500.00 for the Appeal Brief Filing Fee; and
3.   A postcard receipt;

relating to the above application, were deposited as "First Class Mail" with the United States Postal Service, addressed to Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on <u>April 6, 2005</u>.

Date: _____4/6/05_____

Mailer

Date: _____4/6/5_____

Matthew S. Anderson
Reg. No. 39,093

P.O. Drawer 800889
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: *manderson@davismunck.com*

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

	Complete if Known
	Application Number	09/675,699
	Filing Date	09/29/2000
	First Named Inventor	Scott R. McMullan
	Examiner Name	H. Mahmoudi
	Art Unit	2165
	Attorney Docket No.	05-01-004 (UGSC01-05028)

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

METHOD OF PAYMENT (check all that apply)

☑ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify): _____

☑ Deposit Account   Deposit Account Number: 50-0208   Deposit Account Name: Davis Munck, P.C.

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below

☐ Charge fee(s) indicated below, **except for the filing fee**

☑ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17

☑ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
Application Type	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fee ($)**	**Small Entity Fee ($)**	**Fees Paid ($)**
Utility	300	150	500	250	200	100	.
Design	200	100	100	50	130	65	_____
Plant	200	100	300	150	160	80	_____
Reissue	300	150	500	250	600	300	_____
Provisional	200	100	0	0	0	0	_____

2. EXCESS CLAIM FEES

Fee Description	**Fee ($)**	**Small Entity Fee ($)**
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	**Extra Claims**	**Fee ($)**	**Fee Paid ($)**	**Multiple Dependent Claims**	
				Fee ($)	**Fee Paid ($)**
_____ - 20 or HP = _____ x _____ = _____				_____	_____

HP = highest number of total claims paid for, if greater than 20

Indep. Claims	**Extra Claims**	**Fee ($)**	**Fee Paid ($)**
_____ - 3 or HP = _____ x _____ = _____			

HP = highest number of independent claims paid for, if greater than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	**Extra Sheets**	**Number of each additional 50 or fraction thereof**	**Fee ($)**	**Fee Paid ($)**
_____ - 100 = _____ / 50 = _____ (round up to a whole number) x _____ = _____				

4. OTHER FEE(S)

Fees Paid ($)

· Non-English Specification, $130 fee (no small entity discount)   _____

Other: Appeal Brief   500.00

SUBMITTED BY

Signature	[signature]	Registration No. (Attorney/Agent) 39,093	Telephone 972-628-3600
Name (Print/Type)	Matthew S. Anderson		Date April 6, 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : SCOTT R. MCMULLAN, ET AL.

Serial No. : 09/675,699

Filed : September 29, 2000

For : METHOD OF REAL-TIME BUSINESS COLLABORATION

Art Unit : 2165

Examiner : Hassan Mahmoudi

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
. P.O. Box 1450
Alexandria, VA 22313-1450

04/12/2005 MBIZUNES 00000122 09675699
01 FC:1402   500.00 OP

APPEAL BRIEF

Sir:

Applicants herewith respectfully submit that the Examiner's decision of October 20, 2004, finally rejecting Claims 1-18 in the present application, should be reversed, in view of the following arguments and authorities. This Brief is submitted in triplicate on behalf of Appellant for the application identified above. A check is enclosed for the fee for filing a Brief on Appeal. Please charge any additional necessary fees to Deposit Account No. 50-0208.

TABLE OF CONTENTS

APPENDIX A - Text of Claims on Appeal

APPENDIX B - Copy of Informal Drawings

APPENDIX C - Copy of United States Patent Application Publication
    2003/0139975 A1

APPENDIX D - Copy of United States Patent Application 6,412,009

APPENDIX E - Copy of Notice of Appeal previously filed

TABLE OF AUTHORITIES

Real Party in Interest

The real party in interest, and assignee of this case, is UGS Corp. of Plano, Texas (formerly UGS PLM Solutions, Inc. of Plano, Texas).

Related Appeals or Interferences

To the best knowledge and belief of the undersigned attorney, there are none.

Status of Claims

Claims 1-18 are under final rejection, and are each appealed.

Status of Amendments after Final

No amendments were filed after final rejection. A response was filed after final rejection, but this response did not include any amendments.

SUMMARY OF CLAIMED SUBJECT MATTER

The following summary refers to disclosed embodiments and their advantages, but does not delimit any of the claimed inventions.

In General

The present application deals with manipulation of computer network transport protocols so that transport protocols are optimized for purposes of effective network communication, especially with a view to network communication necessary to allow collaborative use of applications by mutually-remote users.

This optimization sometimes involves being sensitive to the way client computers and firewalls operate so that persistent connections, blocking-on-a-read communications, keep-alive communications, stateful communications and the like can be effected through various types of firewalls. This optimization can also involve sensitivity to transport protocols in order to facilitate quick and efficient communication within a network of server computers. In many cases, the best transport protocol for the client computer subsystems is not the same as the best transport protocol for the server computer subsystem.

Given these imperatives, some embodiments of the present invention have a computer program for translating between protocols so that data communications manifest different and optimal transport

protocols at both the server and client ends of the computer system. For example, some firewalls (usually client firewalls) are highly amenable to stateful, HTTP 1.1 keep-alive connections via port 80. Other types of firewalls (usually server firewalls) are configured to allow stateful, persistent communication through non-reserved network ports under other protocols, such as protocols for object serialization. The claimed embodiments can help achieve persistent (and preferably real-time) communication over a computer network that has these and/or other types of protocols by translating data between transport protocols, such that the transport protocols chosen are suited for persistent communication through the relevant firewalls.

Some embodiments of the present invention include a server that has multiple threads and sends data over persistent connections to one or more client computer systems. For example, a server computer system for a real-time, collaborative application (e.g., a collaborative scheduling program) may use

multiple threads for multiple client-collaborator computer systems, and send back data to these client collaborators under HTTP 1.1 protocol in real-time through persistent, stateful, keep-alive connections which were originated through port 80 of each client's firewall.

The present invention deals with computer network architecture and software for facilitating realtime communications. The present invention is thought to be especially helpful in the context of real-time communication in the context of a computer system including one or more firewalls. The most preferred embodiments of the present invention involve real-time application collaboration.

According to one aspect of the present invention, a computer system a first computer network, a first computer subsystem, a second computer subsystem, and a second subsystem firewall. The first computer subsystem includes collaborative application software, with the collaborative application software comprising machine readable instructions for sending application output data over the computer network. The second computer subsystem is structured to receive the application output data. The second subsystem firewall is located in front of the second application subsystem. The second-subsystem firewall is structured to communicate the application output data to the second computer subsystem through a hypertext transfer protocol keep-alive connection that is kept open for the duration of a collaboration.

Support for Independent Claims

Note that, per 37 CFR §41.37, only each of the independent claims are discussed in this section. In the arguments below, however, the dependent claims are also discussed and distinguished from the prior art. The discussion of the claims is for illustrative purposes, and is not intended to affect the scope of the claims.

Figure 1 of the present application is shown on the next page. Note that this application was filed and is being prosecuted with informal drawings. A formal version of Figure 1 was prepared, but has not been filed in this case. It is shown below for the convenience of the Board, but is <u>not</u> formally of record in this case.

Independent claim 1 includes a computer system (100) comprising a first computer network (132); *Fig., 1 and page 11, line 15 - page 12, line 10.* Claim 1 also include a first computer subsystem 102 comprising collaborative application software 150, with the collaborative application software 150 comprising machine readable instructions for sending application output data over the computer network 132; *Fig. 1 and page 12, line 11 - page 20, line 10.*

Independent claim 1 also includes a second computer subsystem 114 structured to receive the application output data; and a second-subsystem firewall 124, located in front of the second application subsystem 114; *Fig 1, page 20, line 11 - page 21, line 12, and page 23, lines 3-8.*

Further, independent claim 1 indicates that the second-subsystem firewall 124 structured to communicate the application output data to the second computer subsystem 114 through a hypertext transfer protocol keep-alive connection that is

kept open for the duration of a collaboration; *Fig 1, page 6, lines 11-18, page 20, line 13 - page 21, line 3, and page 23, lines 3-14.*

Independent claim 15 is drawn to a method of communicating over a computer network, the method comprising the steps of:
generating, by a collaborative application software (150) residing on a server computer (110), an application output communication; sending, over a first computer network (132), the application output communication to a client firewall (124); *Fig 1, page 6, lines 11-16.*

Independent claim 15 also includes communicating the application output communication across the client firewall (124) through a hypertext transfer protocol keep-alive connection; receiving the application output data at a client computer (114); and keeping the hypertext transfer protocol keep-alive connection for the duration of a collaboration, *Fig. 1, page 6, lines 15-18.*

Grounds of Rejection to be Reviewed on Appeal

1. Are Claims 1-18 obvious over Perkowski (US Pub. No. 2003/0139975) in view of Erickson *et al.* (USP 6,412,009)?

ARGUMENT

Stated Grounds of Rejection

The rejections outstanding against the Claims are as follows:

2. Claims 1-18 were rejected as obvious over Perkowski (US Pub. No. 2003/0139975) in view of Erickson et al. (USP 6,412,009). See item3 in the October 20, 2004 Office Action.

Rejections under 35 USC §103

Claims 1-18 were rejected as obvious over Perkowski (US Pub. No. 2003/0139975, hereinafter "Perkowski") in view of Erickson et al. (USP 6,412,009, hereinafter "Erickson").

Legal Standards[1]

The legal standards for an obviousness rejection are referenced in the footnote below.

Analysis of Examiner's Rejection

---

[1]The Supreme Court has explained how to apply §103:

> Under §103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined.
>
> *Graham v. John Deere Co.*, 383 U.S. 1, 148 U.S.P.Q. 459, 467 (1966).*Graham v. John Deere Co.*, 383 U.S. 1, 148 U.S.P.Q. 459, 467 (1966).

Obviousness cannot be inferred from a combination of references without a showing that one of ordinary skill would have been motivated to combine those references:

> When prior art references require selective combination ... to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gained from the invention itself.... Something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination.
>
> *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 5 U.S.P.Q.2d 1434, 1438 (Fed.Cir. 1988).*Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 5 U.S.P.Q.2d 1434, 1438 (Fed.Cir. 1988), *quoting* Such secondary considerations as commercial success, long felt but unsolved needs, failure of others, etc., might be utilized to give light to the circumstances surrounding the origin of the subject matter sought to be patented. As indicia of obviousness or nonobviousness, these inquiries may have relevancy.*Interconnect Planning Corp. v. Feil*, 227 U.S.P.Q. 543 (Fed.Cir. 1985), *and Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick*, 221 U.S.P.Q. 481 (Fed.Cir. 1984).

Perkowski and Erickson are briefly discussed, and then the rejection of each claim is addressed.

Perkowski discloses an Internet-based consumer product information system that allows consumer product information to be transmitted over the Internet using barcode identifiers. Perkowski has a remarkable priority chain of continuations-in-part, which may contribute to its confusing arrangement and numbering of text and figures. Perkowski is not drawn to a data processing system allowing collaborative operation, as is the present application. Perkowski does, however, have some elements with functions or names that are similar to features claimed in the instant application, which will be discussed in detail below. However, these elements to not inter-relate or function as found in the claims.

As may be seen in the detailed analysis below, it appears that Examiner Mahmoudi simply performed a word search on Perkowski's disclosure, and where similar terms have appeared, has made a bare assertion that similar terms must mean that the teachings are identical or at least similar. As will be shown below, this patchwork of citations to various portions of Perkowski's disclosure, without any reasonable attempt to show that Perkowski's features are structurally or functionally similar to the claim limitations, does not appear to even be a good faith attempt to make a *prima facia* obviousness showing.

Erickson is drawn to a method and system for providing a persistent HTTP tunnel. In this limited sense, as discussed below, it is relevant to the claimed "keep-alive" connection; however, as discussed below, there is no motivation to apply these teachings of Erickson to Perkowski.

Claim 1

Claim 1 requires:

A computer system comprising:

a first computer network;

a first computer subsystem comprising collaborative application software,
with the collaborative application software comprising machine
readable instructions for sending application output data over the
computer network;

a second computer subsystem structured to receive the application output
data; and

a second-subsystem firewall, located in front of the second application
subsystem, the second-subsystem firewall structured to communicate
the application output data to the second computer subsystem through
a hypertext transfer protocol keep-alive connection that is kept open
for the duration of a collaboration.

Applicant concedes that some aspects of claim 1 are shown in Perkowski.
For example, Perkowski indeed mentions multiple computer systems, a network,
and even a firewall. However, these elements do not interrelate or function as
claimed, and other elements are completely missing.

Claim 1 requires, in part, "a first computer subsystem comprising
collaborative application software ... for sending application output data over the
computer network ... and ... a second-subsystem firewall, located in front of the
second application subsystem, ... to communicate the application output data to
the second computer subsystem."

These limitations are not taught or suggested by Perkowski. For example, Examiner Mahmoudi uses Perkowski's "Collaborative Replenishment System" to satisfy the claimed "first computer subsystem comprising collaborative application software." However, the only real description of Perkowski's "Collaborative Replenishment Information Subsystem 4," is found in paragraph 185, in relation to figures 2-1 and 2-2. Nothing in this description, these figures, or anywhere else in Perkowski teaches or suggests that the Collaborative Replenishment System sends output data over a network, through a second-subsystem firewall, to a second computer subsystem.

Applicant respectfully notes that Perkowski, in his creatively arranged application, re-uses element numbers in different figures to indicate different objects. For example, the client systems 13 as shown and described with relation to figures 2-1 and 2-2, and described as forming part of the "Collaborative Replenishment System" are connected and used differently than the client system 13 in figure 2A, or in figure 2B1, or in figure 2C, or in figure 3A3', or in figure 3A7, or in figure 4H2, or in figure 5B, or in figure 7, or in many of the other figures. In fact, Perkowski appears to use the identifier "13" for virtually any device in any context.

Applicant further notes that "collaborative" is described in the instant application as "wherein two or more mutually-remote clients concurrently and simultaneously access and control an application (e.g., a word processing application on a remote server machine) over a computer network across one or more firewalls." A "collaborative application" is defined in the specification as "an application capable of concurrently receiving input from and providing output to at least two people at two different computers." Although Perkowski uses the term

"collaborative," nothing in Perkowski or Erickson, below, appear to teach or suggest the use of collaborative application software as described, defined, and claimed in the present application.

As noted above, nothing in Perkowski or Erickson, or any combination of them, appears to teach or suggest the use of collaborative application software as described, defined, and claimed in independent claims 1 and 15 of the present application. As such, these independent claims, and all dependent claims (including claims 6-7 and 17-18) should be allowed over Perkowski and Erickson.

While Perkowski does mention a firewall in several instances, none of these are with regard to the Collaborative Replenishment Information Subsystem. A firewall is discussed with regard to figures 2C and 2D, as part of the client systems of figures 2-1 and 2-2. Paragraph 185 appears to state that the client systems 13 of these figures are themselves a part of the Collaborative Replenishment system, so they cannot a part of the claimed second computer subsystem. For this reason alone, Examiner Mahmoudi should be reversed with regard to Claim 1 and all its direct and indirect dependents.

Examiner Mahmoudi responds to this point, in the final Office Action, that this argument is not persuasive "because 'Collaborative Replenishment System' is not recited in the rejected claim(s)." The Examiner is clearly confused on this point, since the Collaborative Replenishment System is part of Perkowski's disclosure, not the instant application. For a proper rejection, the Examiner Mahmoudi must show that Perkowski discloses or suggests collaborative application software that sends output data over a network, through a second-subsystem firewall, to a second computer subsystem. This feature is not taught or suggested by Perkowski.

Further, claim 1 requires "the second-subsystem firewall structured to communicate the application output data to the second computer subsystem through a hypertext transfer protocol keep-alive connection that is kept open for the duration of a collaboration." Claims 11, 15, and the claims that depend from claim 15 similarly require a keep-alive connection. Nothing in Perkowski teaches or suggests an HTTP keep-alive connection at all. As Examiner Mahmoudi is surely aware (and as discussed in part in Erickson, below), a keep-alive connection has a specific meaning, referring to a specific type of HTTP connection, and has only been supported in later versions of the HTTP specifications. Nothing in Perkowski teaches or suggests this type of connection (which is not believed to have even existed at the time of filing of earlier applications in Perkowski's chain-of-priority), and the term "alive" is does not even appear in Perkowski. The final Office Action references paragraph 178 for this limitation, apparently for the "dedicated Internet connection" language, but a dedicated physical Internet connection is not at all the same as an HTTP keep-alive connection.[2] Erickson does teach a keep-alive connection, but it does not function as claimed.

Claim 1 further requires, "the second-subsystem firewall structured to communicate the application output data to the second computer subsystem through a hypertext transfer protocol keep-alive connection that is kept open for the duration of a collaboration." Nothing in Perkowski discusses a keep-alive connection, and nothing in Perkowski, Erickson, or a combination of them discusses an HTTP keep-alive connection that is kept open for the duration of a

---

[2]"Also, the WebDox™ Server is provided with a dedicated Internet connection (i.e. ISDN or better) to the Internet infrastructure." Perkowski, from Paragraph 178.

collaboration, for the simple reason that none of the cited art discusses this type of collaboration at all.

Claim 2

Claim 2 requires: "the computer system further comprises communication software comprising machine readable instructions for opening a first-subsystem thread in the second computer subsystem for receiving the application output data."

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

Nothing in Perkowski teaches this limitation. Certainly paragraph 163, cited by Examiner Mahmoudi, does not – there is no teaching of a first-subsystem thread in the second computer subsystem, and no teaching that any such tread receives application output data from the first computer subsystem (which, to meet the "collaborative application software" limitation of claim 1, can only arguably be Perkowski's "collaborative replenishment system."). The only reference to a thread at all appears in paragraph 764, and this does not meet the claim limitation.

Claim 3

]   Claim 3 requires "the second computer subsystem comprises a second-subsystem socket structured to receive the application output data; and the communication software further comprises machine readable instructions for causing the second-subsystem socket to block on a read."

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

With regard to claims 3, 10, 12, and 16, nothing in Perkowski or Erickson teaches or suggests a socket that blocks on a read. "Block on a read" generally means that when a "read" process is used on a socket, any other threads or processes are "blocked" from accessing the socket. This has nothing to do with "performing a search," and the undersigned has remained mystified as to why Examiner Mahmoudi indicates that "block on a read" reads on "performing a search." Examiner Mahmoudi, in the final Office Action, responds that he reads "blocking on a read" as "carrying out a search" in Perkowski's paragraph 206. Nothing in the paragraph teaches or suggests anything about blocking on a read, nor that "carrying out a search" is functionally equivalent to blocking on a read. Examiner Mahmoudi has been respectfully and repeatedly requested to provide documentary support for his unusual interpretation, and has been unable or unwilling to do so.

Claim 4

Claim 4 requires: "the communication software further comprises instructions causing the first-subsystem thread to sleep." This limitation, particularly in the claimed context of the first-subsystem thread in the second computer subsystem that receives the application output data, is not taught or suggest by any cited art.

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

Examiner Mahmoudi's final rejection cites Perkowski, paragraph 233, "where 'sleep' is read on 'idle moment'". The relevant sentence in Perkowski refers to kiosks in a retail store, and reads in its entirety:

> It is understood, however, that in many application [*sic*], in which advertisements, prices and specials, notices and the like are to be displayed on the kiosks during idle moments (i.e. when consumers are not scanning bar coded products for consumer product related information access and display), there will be a need to use a more robust electronic messaging and http server solutions [*sic*] on the retailer's network information server **84**.

This rejection appears to equate "causing the first-subsystem thread to sleep" with an entire client system being in an "idle" mode because it is not actively being used. These are entirely different concepts, and as nothing in Perkowski teaches or suggests the operation of threads at all, certainly nothing teaches or suggests putting a thread to sleep. Nothing in Perkowski or Erickson teaches or suggests

that communication software includes instructions to put a specific thread to sleep, where the specific thread has the function and interrelation with other elements as claimed.

Claim 5

Claim 5 requires: "the collaborative application software sends the application output data as a stateful communication." This limitation is not taught or suggested by any art of record.

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

Examiner Mahmoudi's final rejection relies on Perkowski, paragraph 340, "where 'stateful' is read on 'reflecting the state of the client and the server.'" Perkowski's paragraph 340 refers to a "request response model" between a client subsystem 13 and a Java Web Server 11, where the "request and the corresponding response reflect the state of the client and the server at the time of the request."

Nothing in this passage teaches anything at all about the application output date of collaborative application software, as claimed. Further, nothing in any cited art suggests that any collaborative application software on any system sends any output data as a stateful communication.

Claim 6

Claim 6 requires: "the application output data is structured and arranged according to an HTTP 1.1 protocol."

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

While Perkowski and Erickson both discuss HTTP transactions, and Erickson discusses HTTP 1.1 transactions, none of them are in the claimed context of a first computer subsystem comprising collaborative application software, with the collaborative application software comprising machine readable instructions for sending application output data over the computer network, where the application output data is structured and arranged according to an HTTP 1.1 protocol.

Claim 7

Claim 7 requires: "the second-subsystem firewall comprises a port 80; and the application output data is communicated across the second-subsystem firewall through a connection originated through port 80."

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

While Erickson does discuss a firewall with a port 80, there is no teaching in any cited art with regard to the port being in a second-subsystem firewall or passing the application output data as claimed, and as discussed more fully with regard to claim 1.

Claim 8

Claim 8 requires: "wherein the first computer subsystem comprises: a server computer; a Web server computer, and a second computer network structured to allow data communication between the server computer and the Web server computer."

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim.

Here, the final Office Action indicates that the claimed first computer subsystem is satisfied by server computer 202, network bridge server computer 133 (also described by Perkowski as a web server), and the Internet infrastructure, all shown in Perkowski's figure 2C. None of these systems include the claimed "collaborative application software, with the collaborative application software comprising machine readable instructions for sending application output data over the computer network," as required in the first computer subsystem by claim 1.

Applicant notes, as discussed above with relation to claim 1, that the client system 13 shown in figure 2C does not appear to be the same as the client system 13 described with relation to figures 2-1 and 2-2.

Claim 9

Claim 9 requires: "the server computer comprises at least a portion of the collaborative applications software; and the Web server computer is structured to receive the application output data from the server computer over the second computer network and to send the application output data to the second computer subsystem over the first computer network." This limitation is not taught or suggested by any art of record.

The arguments regarding the limitations of parent claims 1 and 8, above, are incorporated herein with regard to this claim.

Perkowski's Collaborative Replenishment System, the only element that arguably has any collaborative application software, does not appear to send data to a Web server over a second computer network and from there to a second computer subsystem over a first computer network.

Claim 10

Claim 10 requires: "the Web server computer comprises a Web server socket structured to receive the application output data from the server computer over the second computer network; and the communication software further comprises machine readable instructions for causing the Web server socket to block on a read." This limitation is not taught or suggested by any art of record.

The arguments regarding the limitations of parent claims 1, 8 and 9, above, are incorporated herein with regard to this claim.

Further, as described above, nothing in Perkowski or Erickson teaches or suggests a socket that blocks on a read. "Block on a read" generally means that when a "read" process is used on a socket, any other threads or processes are "blocked" from accessing the socket. This has nothing to do with "performing a search," and the undersigned has remained mystified as to why Examiner Mahmoudi indicates that "block on a read" reads on "performing a search." Examiner Mahmoudi, in the final Office Action, responds that he reads "blocking on a read" as "carrying out a search" in Perkowski's paragraph 206. Nothing in the paragraph teaches or suggests anything about blocking on a read, nor that "carrying out a search" is functionally equivalent to blocking on a read. Examiner Mahmoudi has been respectfully and repeatedly requested to provide documentary support for his unusual interpretation, and has been unable or unwilling to do so.

Claim 11

Claim 11 requires: "a third computer subsystem structured to receive the application output data; and a third-subsystem firewall, located in front of the third computer subsystem the third-subsystem firewall structured to communicate the application output data to the third computer subsystem through a hypertext transfer protocol keep-alive connection." This limitation is not taught or suggested by any art of record.

The arguments regarding the limitations of parent claim 1, above, are incorporated herein with regard to this claim. In particular, none of the cited art or any combination of them describe a third computer subsystem that is connected and functions as described, including receiving the application output date through an HTTP keep-alive connection.

Claim 12

Claim 12 requires: "the third computer subsystem comprises a third-subsystem socket structured to receive the application output data; and the communication software further comprises machine readable instructions for causing the third-subsystem socket to block on a read." This limitation is not taught or suggested by any art of record.

The arguments regarding the limitations of parent claims 1 and 11, above, are incorporated herein with regard to this claim.

Further, as described above, nothing in Perkowski or Erickson teaches or suggests a socket that blocks on a read. "Block on a read" generally means that when a "read" process is used on a socket, any other threads or processes are "blocked" from accessing the socket. This has nothing to do with "performing a search," and the undersigned has remained mystified as to why Examiner Mahmoudi indicates that "block on a read" reads on "performing a search." Examiner Mahmoudi, in the final Office Action, responds that he reads "blocking on a read" as "carrying out a search" in Perkowski's paragraph 206. Nothing in the paragraph teaches or suggests anything about blocking on a read, nor that "carrying out a search" is functionally equivalent to blocking on a read. Examiner Mahmoudi has been respectfully and repeatedly requested to provide documentary support for his unusual interpretation, and has been unable or unwilling to do so.

Claim 13

Claim 13 requires: "wherein communication between the first computer subsystem, the second computer subsystem and the third computer subsystem is in real-time."

The arguments regarding the limitations of parent claims 1 and 11, above, are incorporated herein with regard to this claim.

Certainly Perkowski contemplates that some computer communications are in real-time. This limitation is particularly important in the context of the system of the parent claims; the parent claim limitations are not met by any cited art.

Claim 14

Claim 14 requires: "wherein the collaborative application software comprises at least one of the following functions: a word processor, a task scheduling tool, a graphics program, a presentation program, a spreadsheet, a game, a music studio." This feature is not taught or suggested by the art of record.

The arguments regarding the limitations of parent claims 1 and 11, above, are incorporated herein with regard to this claim.

While Examiner Mahmoudi correctly notes that Perkowski makes reference to word processing and graphics program, this reference is not in relation to the claimed <u>collaborative application software</u>.

Claim 15

Claim 15 requires:

> A method of communicating over a computer network, the method comprising the steps of:
>
> generating, by a collaborative application software residing on a server computer, an application output communication;
>
> sending, over a first computer network, the application output communication to a client firewall;
>
> communicating the application output communication across the client firewall through a hypertext transfer protocol keep-alive connection;
>
> receiving the application output data at a client computer; and
>
> keeping the hypertext transfer protocol keep-alive connection for the duration of a collaboration.

This combination of features and limitation is not taught or suggested by any combination of the art of record.

The arguments made above with regard to claim 1 apply similarly to claim 15, and are incorporated herein by reference.

In particular, the limitations of claim 15 are not taught or suggested by Perkowski. In the case of this claim, Examiner Mahmoudi makes no attempt at all to address the claimed "collaborative application software residing on a server computer." As noted above, nothing in Perkowski or Erickson, or any combination of them, appears to teach or suggest the use of collaborative application software as described, defined, and claimed in independent claim 15 of

the present application. As such, this independent claim, and all dependent claims (including claims 17-18) should be allowed over Perkowski and Erickson.

While Perkowski does mention a firewall in several instances, none of these are with regard to a server system having collaborative application software.

Further, claim 15 requires "communicating the application output communication across the client firewall through a hypertext transfer protocol keep-alive connection; ... and keeping the hypertext transfer protocol keep-alive connection for the duration of a collaboration." Claims 15, and the claims that depend from claim 15 similarly require a keep-alive connection. Nothing in Perkowski teaches or suggests an HTTP keep-alive connection at all. The final Office Action references paragraph 178 for this limitation, apparently for the "dedicated Internet connection" language, but a dedicated physical Internet connection is not at all the same as an HTTP keep-alive connection.[3] Erickson does teach a keep-alive connection, but it does not function as claimed. Nothing in Perkowski discusses a keep-alive connection, and nothing in Perkowski, Erickson, or a combination of them discusses an HTTP keep-alive connection that is kept open for the duration of a collaboration, for the simple reason that none of the cited art discusses this type of collaboration at all.

---

[3]"Also, the WebDox™ Server is provided with a dedicated Internet connection (i.e. ISDN or better) to the Internet infrastructure." Perkowski, from Paragraph 178.

Claim 16

Claim 16 requires: "the client computer blocks on a read when waiting for and receiving the application output data." This feature is not taught or suggested by the art of record.

The arguments regarding the limitations of parent claim 15, above, are incorporated herein with regard to this claim.

Further, as described above, nothing in Perkowski or Erickson teaches or suggests a socket that blocks on a read. "Block on a read" generally means that when a "read" process is used on a socket, any other threads or processes are "blocked" from accessing the socket. This has nothing to do with "performing a search," and the undersigned has remained mystified as to why Examiner Mahmoudi indicates that "block on a read" reads on "performing a search." Examiner Mahmoudi, in the final Office Action, responds that he reads "blocking on a read" as "carrying out a search" in Perkowski's paragraph 206. Nothing in the paragraph teaches or suggests anything about blocking on a read, nor that "carrying out a search" is functionally equivalent to blocking on a read. Examiner Mahmoudi has been respectfully and repeatedly requested to provide documentary support for his unusual interpretation, and has been unable or unwilling to do so.

Claim 17

Claim 17 requires: "originating a connection across the client firewall through a port 80 of client firewall."

The arguments regarding the limitations of parent claim 15, above, are incorporated herein with regard to this claim.

While Erickson does discuss a firewall with a port 80, there is no teaching in any cited art with regard to several of the additional limitations of claim 15, discussed above.

Claim 18

Claim 18 requires: "the application output data is sent, at the sending step, as a plurality of data packets structured and arranged according to HTTP 1.1."

The arguments regarding the limitations of parent claim 15, above, are incorporated herein with regard to this claim.

While Perkowski and Erickson both discuss HTTP transactions, and Erickson discusses HTTP 1.1 transactions, none of them are in the claimed context of the method of parent claim 15, where the application output communication is structured and arranged according to an HTTP 1.1 protocol.

Motivation to Combine or Modify[4]

Erickson is cited in addition to Perkowski apparently for the sole teaching of a "keep-alive connection." Claim 1 requires "the second-subsystem firewall structured to communicate the application output data to the second computer subsystem <u>through a hypertext transfer protocol keep-alive connection that is kept open for the duration of a collaboration</u>." Claims 11, 15, and the claims that depend from claim 15 similarly require a keep-alive connection. As the Examiner concedes, nothing in Perkowski teaches or suggests an HTTP keep-alive connection that is kept open for the duration of a collaboration. Erickson discusses a "keep-alive" function for maintaining a persistent HTTP tunnel for a connection-oriented protocol between a client and a web server.

---

[4]Where an obviousness rejection is based on a combination of references, the Examiner must show that one of ordinary skill would have been motivated to combine those references. *See In re Nilssen*, 7 USPQ2d 1500 (Fed.Cir. 1988).*In re Nilssen*, 7 USPQ2d 1500 (Fed.Cir. 1988); *Panduit Corp. v. Dennison Mfg. Co.*, 1 USPQ2d 1593, 1597 (Fed.Cir. 1987).*Panduit Corp. v. Dennison Mfg. Co.*, 1 USPQ2d 1593, 1597 (Fed.Cir. 1987); *ACS Hospital Systems v. Montefiore Hospital*, 220 USPQ 929 (Fed.Cir. 1984). *ACS Hospital Systems v. Montefiore Hospital*, 220 USPQ 929 (Fed.Cir. 1984). "When prior art references require selective combination ... to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight gained from the invention itself.... Something in the prior art as a whole must suggest the desirability, and thus the obviousness, of making the combination." *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 5 USPQ2d 1434, 1438 (Fed.Cir. 1988).*Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 5 USPQ2d 1434, 1438 (Fed.Cir. 1988), *quoting Interconnect Planning Corp. v. Feil*, 227 USPQ 543 (Fed.Cir. 1985), *and Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick*, 221 USPQ 481 (Fed.Cir. 1984). "While *[a reference]* may be capable of being modified to run the way *[the applicant's]* apparatus is claimed, there must be a suggestion or motivation in the reference to do so. See In re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984) ("The mere fact that the prior art could be so modified would not have made the modification obvious unless the prior art suggested the desirability of the modification."). *In re*

Examiner Mahmoudi states that it would have been obvious to modify Perkowski to include the "keep-alive" connection of Erickson that is kept open for the duration of a collaboration because it "would enable the system to keep the connection active/alive even during periods of inactivity." The Examiner was respectfully requested to show where Perkowski teaches or suggests that keeping a connection active is advantageous or even considered, and was unable or unwilling to do so. There appears to be no such motivation discussed in the references themselves, and nothing to indicate that Erickson's approach would be advantageous or even operable in Perkowski's system. If a basis for the alleged motivation cannot be shown in the art, then claims 1 and 15, and all of their direct and indirect dependents, should be allowed over Perkowski and Erickson.

In particular, it should be noted that Claims 1 and 15 specifically claim that the keep-alive connection is <u>for the duration of a collaboration</u>, and has nothing at all to do with keeping the connection alive even during periods of inactivity, the motivation stated by Examiner Mahmoudi. In fact, it appears that the Examiner's "motivation" is contrary to the claim language itself.

Examiner Mahmoudi indicated in the final Office Action that he is relying on knowledge generally available to one of ordinary skill in the art, and apparently concedes that no such motivation can be found in the references themselves. The Examiner's stated motivational advantage, to "enable the system to keep the connection active/alive even during periods of inactivity," is not taught or suggested by Perkowski to be desirable at all, and nothing in Erickson indicates that such an ability would be advantageous in Perkowski's system. There is no teaching or suggestion at all - other than the Examiner's bare assertion -- that such

an ability is generally desirable, or that such an ability would be advantageous or functional in Perkowski's system. Accordingly, the Examiner was requested to provide documentary evidence or an affidavit establishing what precise knowledge he finds to be "generally available to one of ordinary skill in the art," and where the stated motivation can be found, as required by MPEP § 2144.03. Examiner Mahmoudi was unable or unwilling to fulfill this requirement of the MPEP.
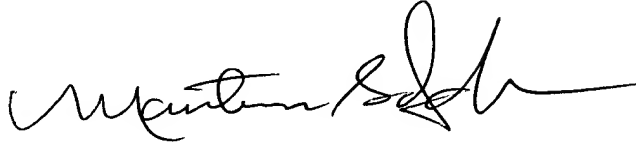
Grouping of Claims

The claims on appeal <u>do not</u> stand or fall together, as may be seen from the arguments set forth below. Each claim has been argued separately under a separate subheading, and each claim should be considered separately. While the applicant recognizes that a formal statement regarding the grouping of claims is no longer required, each claim should be considered separately; or at the very least each claim which is argued separately in the preceding sections of this brief should be considered separately. <u>Argument:</u> The fact that the claims use different formulations (as detailed above) and/or have been argued separately, shows that, if their patentability is not considered separately, any adverse decision would show that the limitations of some claims had been unfairly ignored.

REQUESTED RELIEF

The Board is respectfully requested to reverse the outstanding rejections and return this application to the Examiner for allowance.

Respectfully submitted,

Matthew S. Anderson, Reg.No. 39,093, for:

DAVIS MUNCK PC
13155 Noel Rd., Suite 900
Dallas, TX 75240

Attorney for Applicant

April 6, 2005

In the United States Patent and Trademark Office

In re application of	: Scott R. McMullan, et al.
Serial No.	: 09/675,699
Filed	: September 29, 2000
For	: Method of Real-Time Business Collaboration
Art Unit	: 2165
Examiner	: Hassan Mahmoudi

APPENDIX A -

Text of Claims on Appeal

1. (Original) A computer system comprising:

    a first computer network;

    a first computer subsystem comprising collaborative application software, with
    the collaborative application software comprising machine readable
    instructions for sending application output data over the computer network;

    a second computer subsystem structured to receive the application output data;
    and

    a second-subsystem firewall, located in front of the second application
    subsystem, the second-subsystem firewall structured to communicate the

application output data to the second computer subsystem through a hypertext transfer protocol keep-alive connection that is kept open for the duration of a collaboration.

2.   (Original) The computer system of claim 1 wherein the computer system further comprises communication software comprising machine readable instructions for opening a first-subsystem thread in the second computer subsystem for receiving the application output data.

3.   (Original) The computer system of claim 2 wherein:

the second computer subsystem comprises a second-subsystem socket structured to receive the application output data; and

the communication software further comprises machine readable instructions for causing the second-subsystem socket to block on a read.

4.   (Original) The system of claim 3 wherein the communication software further comprises instructions causing the first-subsystem thread to sleep.

5.   (Original) The system of claim 1 wherein the collaborative application software sends the application output data as a stateful communication.

6. (Original) The system of claim 5, wherein the application output data is structured and arranged according to an HTTP 1.1 protocol.

7.   (Original) The system of claim 6 wherein:

the second-subsystem firewall comprises a port 80; and

the application output data is communicated across the second-subsystem firewall through a connection originated through port 80.

8.   (Original) The system of claim 1 wherein the first computer subsystem comprises:

a server computer;

a Web server computer, and

a second computer network structured to allow data communication between the server computer and the Web server computer.

9. (Original) The system of claim 8 wherein:

the server computer comprises at least a portion of the collaborative applications software; and

the Web server computer is structured to receive the application output data from the server computer over the second computer network and to send the application output data to the second computer subsystem over the first computer network.

10. (Original) The system of claim 9 wherein:

the Web server computer comprises a Web server socket structured to receive the application output data from the server computer over the second computer network; and

the communication software further comprises machine readable instructions for causing the Web server socket to block on a read.

11. (Original) The system of claim 1, further comprising:

a third computer subsystem structured to receive the application output data; and

a third-subsystem firewall, located in front of the third computer subsystem the third-subsystem firewall structured to communicate the application output data to the third computer subsystem through a hypertext transfer protocol keep-alive connection.

12.   (Original) The computer system of claim 11 wherein:

the third computer subsystem comprises a third-subsystem socket structured to

receive the application output data; and

the communication software further comprises machine readable instructions for

causing the third-subsystem socket to block on a read.

13.   (Original) The system of claim 11 wherein communication between the first computer subsystem, the second computer subsystem and the third computer subsystem is in real-time.

14.   (Original) The system of claim 11 wherein the collaborative application software comprises at least one of the following functions: a word processor, a task scheduling tool, a graphics program, a presentation program, a spreadsheet, a game, a music studio.

15.   (Original) A method of communicating over a computer network, the method comprising the steps of:

    generating, by a collaborative application software residing on a server computer, an application output communication;

    sending, over a first computer network, the application output communication to a client firewall;

    communicating the application output communication across the client firewall through a hypertext transfer protocol keep-alive connection;

    receiving the application output data at a client computer; and

    keeping the hypertext transfer protocol keep-alive connection for the duration of a collaboration.

16.   (Original) The method of claim 15 wherein the client computer blocks on a read when waiting for and receiving the application output data.

17.   (Original) The method of claim 15, further comprising the step of originating a connection across the client firewall through a port 80 of client firewall.
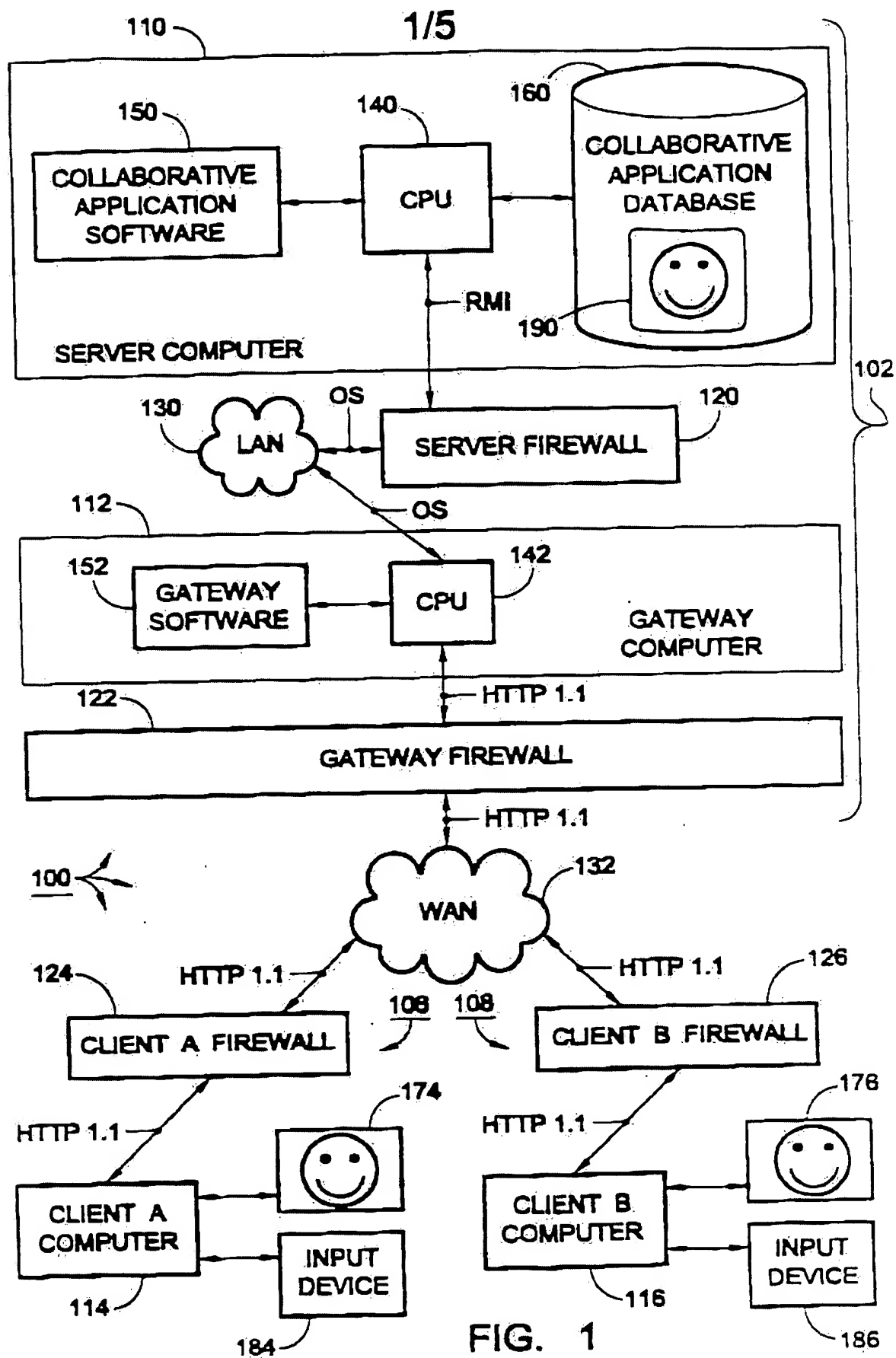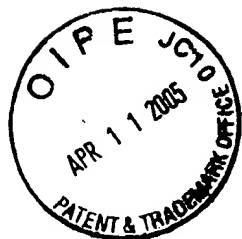
18. (Original) The method of claim 15 wherein the application output data is sent, at the sending step, as a plurality of data packets structured and arranged according to HTTP 1.1.
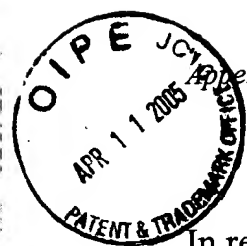
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : SCOTT R. MCMULLAN, ET AL.

Serial No. : 09/675,699

Filed : September 29, 2000

For : METHOD OF REAL-TIME BUSINESS
   COLLABORATION

Art Unit : 2165

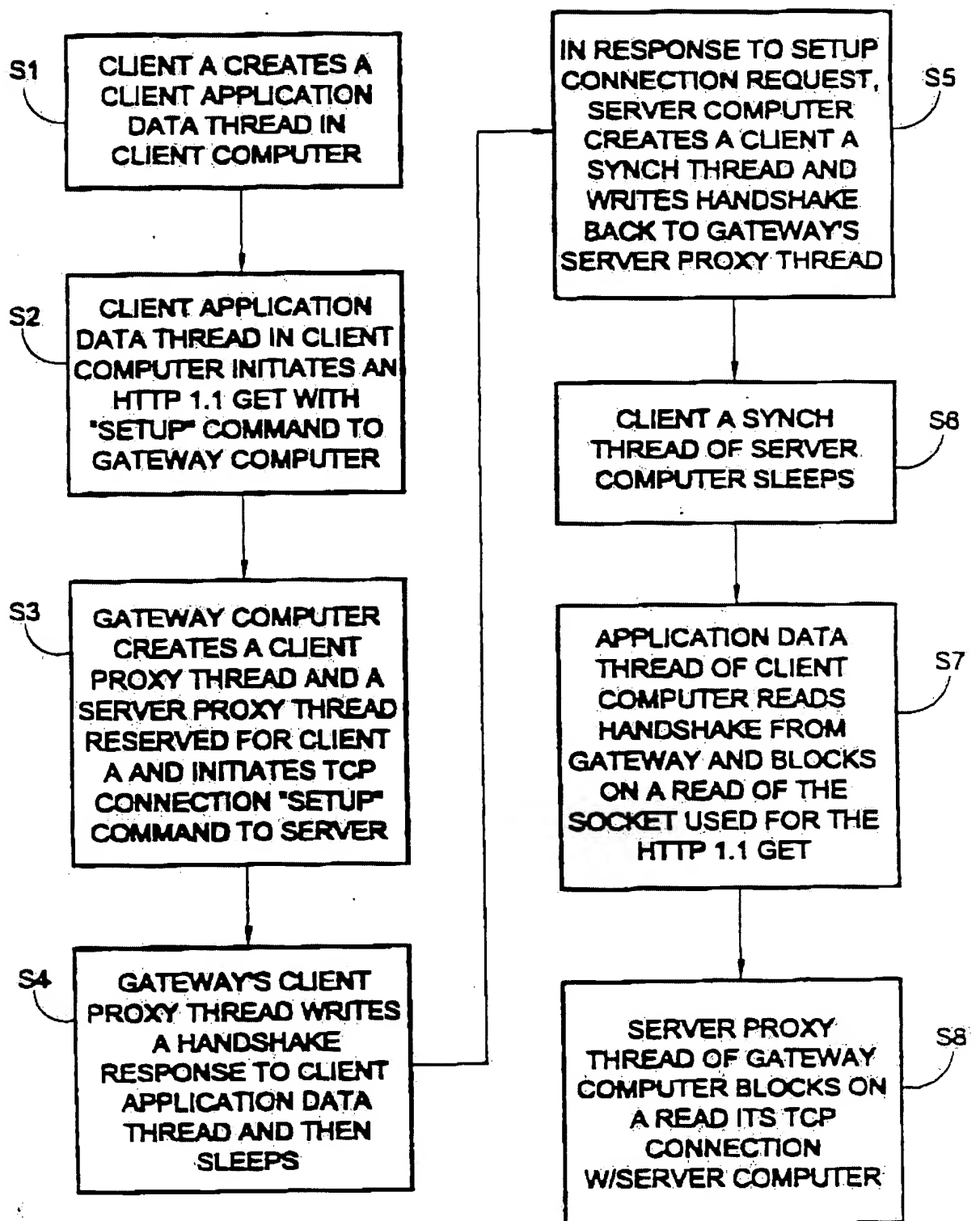Examiner : Hassan Mahmoudi

APPENDIX B -

Copy of Formal Drawings
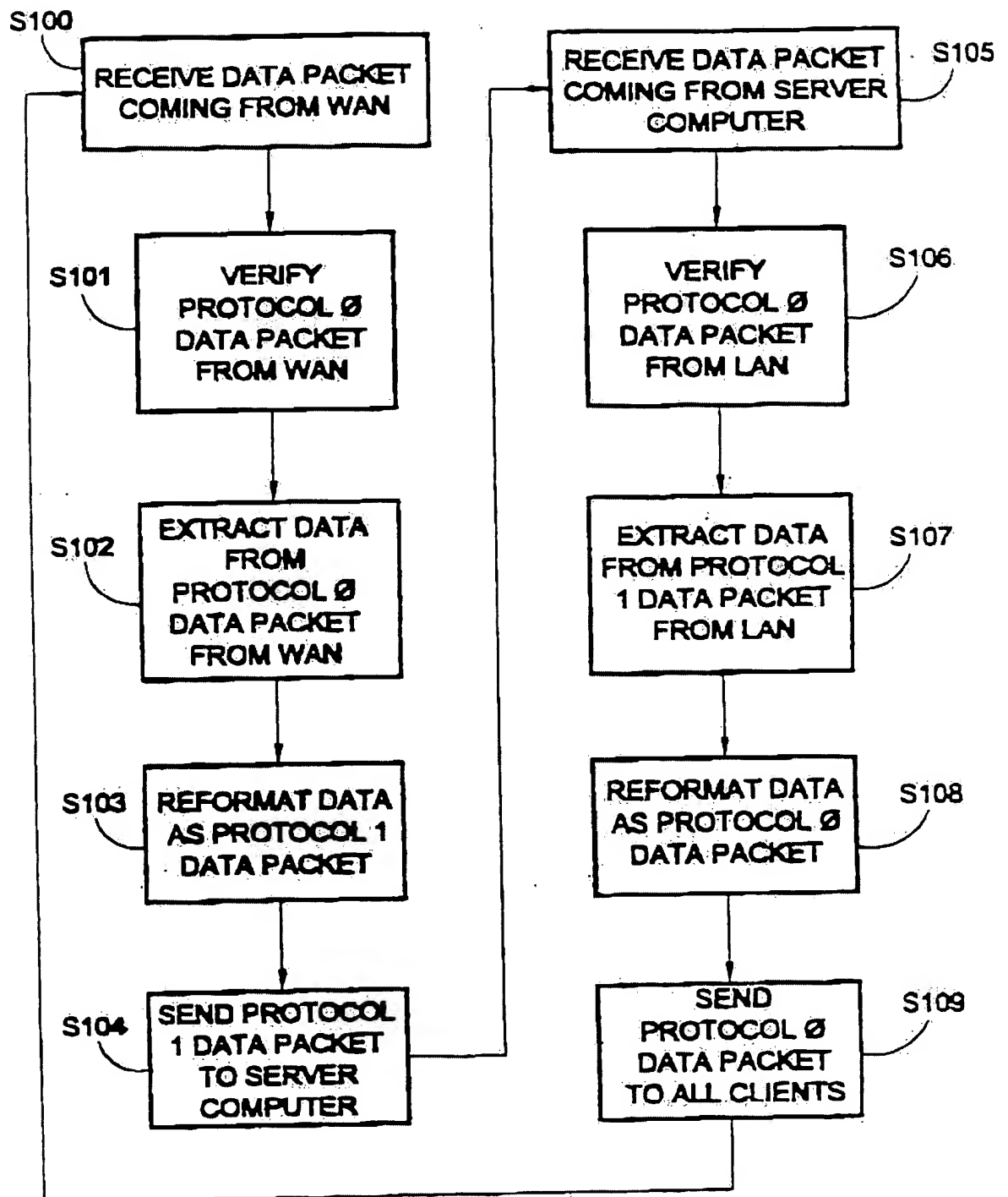
FIG. 1

```
┌─────────────────────┐
│ CLIENT A CREATES A  │
S1│ CLIENT APPLICATION  │
│ DATA THREAD IN      │
│ CLIENT COMPUTER     │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ CLIENT APPLICATION  │
│ DATA THREAD IN CLIENT│
S2│ COMPUTER INITIATES AN│
│ HTTP 1.1 GET WITH   │
│ "SETUP" COMMAND TO  │
│ GATEWAY COMPUTER    │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ GATEWAY COMPUTER    │
│ CREATES A CLIENT    │
S3│ PROXY THREAD AND A  │
│ SERVER PROXY THREAD │
│ RESERVED FOR CLIENT │
│ A AND INITIATES TCP │
│ CONNECTION "SETUP"  │
│ COMMAND TO SERVER   │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ GATEWAY'S CLIENT    │
│ PROXY THREAD WRITES │
S4│ A HANDSHAKE         │
│ RESPONSE TO CLIENT  │
│ APPLICATION DATA    │
│ THREAD AND THEN     │
│ SLEEPS              │
└─────────────────────┘
```

```
┌─────────────────────┐
│ IN RESPONSE TO SETUP│
│ CONNECTION REQUEST, │
│ SERVER COMPUTER     │S5
│ CREATES A CLIENT A  │
│ SYNCH THREAD AND    │
│ WRITES HANDSHAKE    │
│ BACK TO GATEWAY'S   │
│ SERVER PROXY THREAD │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ CLIENT A SYNCH      │
│ THREAD OF SERVER    │S6
│ COMPUTER SLEEPS     │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ APPLICATION DATA    │
│ THREAD OF CLIENT    │S7
│ COMPUTER READS      │
│ HANDSHAKE FROM      │
│ GATEWAY AND BLOCKS  │
│ ON A READ OF THE    │
│ SOCKET USED FOR THE │
│ HTTP 1.1 GET        │
└─────────────────────┘
           │
           ▼
┌─────────────────────┐
│ SERVER PROXY        │
│ THREAD OF GATEWAY   │S8
│ COMPUTER BLOCKS ON  │
│ A READ ITS TCP      │
│ CONNECTION          │
│ W/SERVER COMPUTER   │
└─────────────────────┘
```
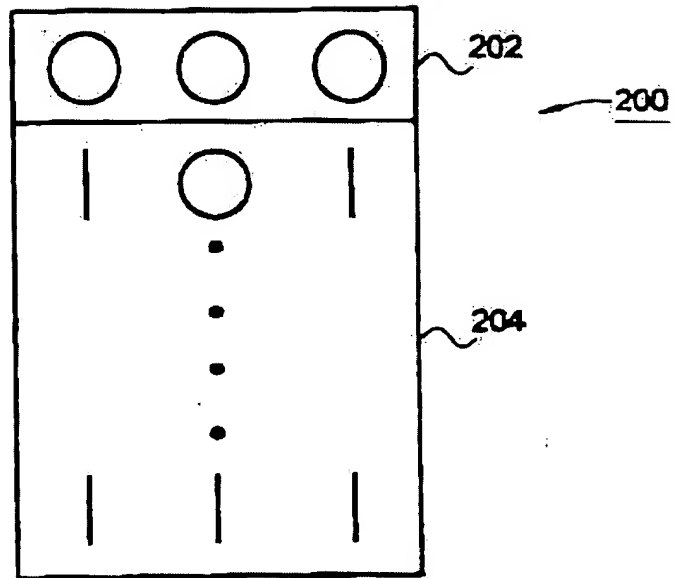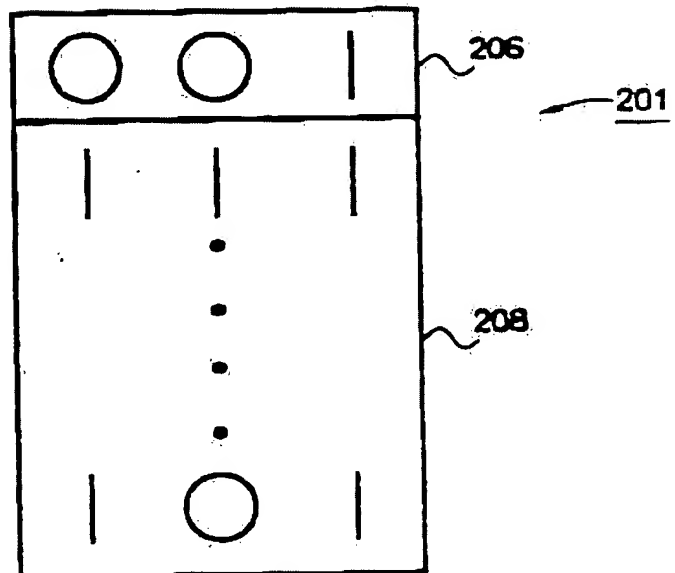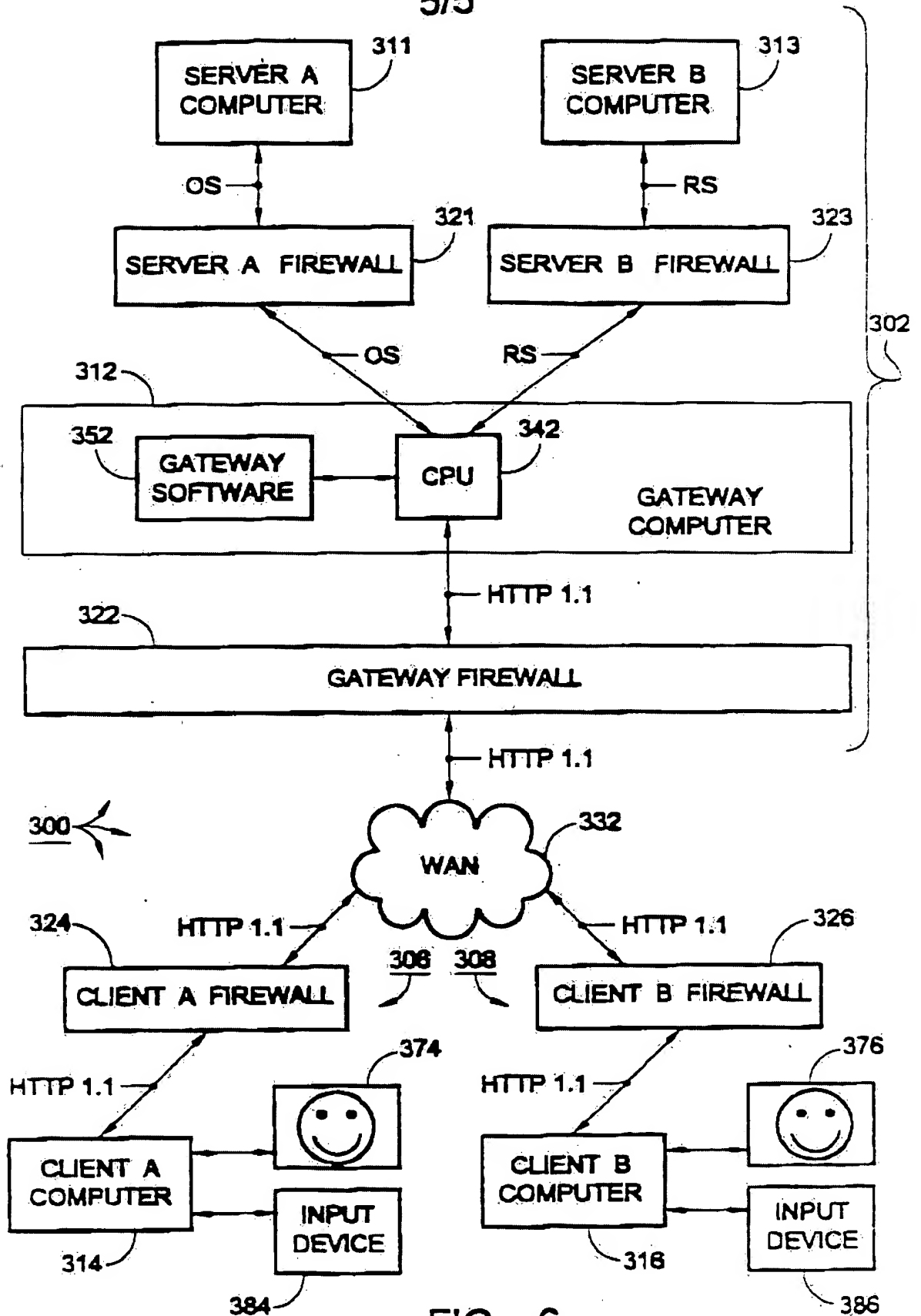
FIG. 2

FIG. 3

FIG. 4

FIG. 5

5/5

FIG. 6

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : SCOTT R. MCMULLAN, ET AL.

Serial No. : 09/675,699

Filed : September 29, 2000

For : METHOD OF REAL-TIME BUSINESS COLLABORATION

Art Unit : 2165

Examiner : Hassan Mahmoudi

APPENDIX C -

Copy of United States Patent Application Publication 2003/0139975 A1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of   : SCOTT R. MCMULLAN, ET AL.

Serial No.   : 09/675,699

Filed   : September 29, 2000

For   : METHOD OF REAL-TIME BUSINESS
                                         COLLABORATION

Art Unit   : 2165

Examiner   : Hassan Mahmoudi

APPENDIX D -

Copy of United States Patent Application 6,412,009

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of : SCOTT R. MCMULLAN, ET AL.

Serial No. : 09/675,699

Filed : September 29, 2000

For : METHOD OF REAL-TIME BUSINESS
      COLLABORATION

Art Unit : 2165

Examiner : Hassan Mahmoudi

APPENDIX E -

Copy of Notice of Appeal Previously Filed

NOTICE OF APPEAL FROM THE EXAMINER TO THE BOARD OF PATENT APPEALS AND INTERFERENCES	Docket Number (Optional) 05-01-004

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" on _March 1, 2005_ Signature _Nancy J. Todd_ Typed or printed Name _Nancy J. Todd_	In re Application of **Scott R. McMullan, et al.**

Application Number 09/675,699	Filed 9/29/2000

For **METHOD OF REAL-TIME BUSINESS COLLABORATION**

Group Art Unit 2165	Examiner H. Mahmoudi

Applicant hereby **appeals** to the Board of Patent Appeals and Interferences from the last decision of the examiner.

The fee for this Notice of Appeal is (37 CFR 1.17(b)) $ _500.00_ .

[ ] Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee shown above is reduced by half, and the resulting fee is: $ _____ .

[X] A check in the amount of the fee is enclosed.

[ ] Payment by credit card. Form PTO-2038 is attached.

[ ] The Commissioner has already been authorized to charge any fees in this application to a Deposit Account. I have enclosed a duplicate copy of this sheet.

[X] The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. _50-0208_. I have enclosed a duplicate copy of this sheet.

[X] A petition for an extension of time under 37 CFR 1.136(a) (PTO/SB/22) is enclosed.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

[ ] applicant/inventor.

[ ] assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

[X] attorney or agent of record.

[ ] attorney or agent acting under 37 CFR 1.34(a). Registration number if acting under 37 CFR 1.34(a). _____ .

_____
Signature

Matthew S. Anderson, Reg. 39,093
Typed or printed name

March 1, 2005
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

[ ] *Total of _____ forms are submitted.